Key Success Factors in Control System Software Architecture

Veli-Pekka Eloranta, Johannes Koskinen, Marko Leppänen
(Tampere University of Technology)

The amount of the control system software used in the mobile work machines has increased radically during the last ten years. The software enables development of new features and further automation. Consequently, the productivity of the work machines increases, bringing cost savings.

However, the development of control system software is challenging because they have special characteristics, such as close dependency to hardware, strict real-time requirements, distribution, and long life-cycle of the products. In addition, standards and regulations dictate how functional safety should be taken into account.

In software engineering, architecture is often seen as the key element of the system enabling high quality design and implementation. Good software architecture helps the developer to easily maintain and further develop the code through the life cycle of the system. Therefore, in this paper we clarify which quality properties and constraints are important now and in the future from the point of view of the software architects of the work machine control systems. The study is based on the questionnaire targeted to the experts of the field and various interview surveys we have carried out during several years in the leading international work machine manufacturers.

Introduction

In a recent issue of IEEE Software, Frank Buschmann et al. [1] present a recent study [2] by Ameller et al. where the researchers conclude that non-functional requirements (NFRs) are rarely the driving factors for designing and implementing software. The study suggests that the customers see the value mostly in functionality and quality attributes are mainly software architects concern. They see this as a surprising result; NFRs are seen as key marketing points by many people, but it seems that the architects are the source of the non-functional requirements.

The software in mobile machine control (MMC) systems domain is usually considered as heavily architecturally driven and aims at building a system where integration of several subsystems is supported. This is due the fact that the machine control systems companies rely a lot on subcontracting. Eklund & Bosch [3] define their mass-produced embedded systems software as having a close dependency to hardware, focus on manufacturing and strong supplier involvement. The systems also typically exhibit safety-critical functionality. Furthermore, these systems typically have life-cycle of tens of years as after years of design the machine is manufactured for several years and after that support for 10-15 years might be promised to the customer. In addition, the machine might contain legacy parts from older generations. In many cases, the machine control systems have strict real-time requirements as they have to react quickly to various environmental stimuli. On architecture level, this fast response time is usually achieved by distributing the system. Thus, this domain is clearly differentiated from the other software engineering domains.

In the original study [2], only one of the 13 respondents was from the aerospace industry, which is a part of the machine control system domain. We decided to explore if the similar results hold for MMC domain in general. Traditionally it is regarded that the companies in MMC domain hold software quality in high regard as in this domain software has long life-cycles and the systems are used in environments where safety and high availability are usually required. The situation is, however, changing as control system software is becoming more important factor and becoming a differentiating point from the competitors. This means time-to-market has to be shortened and release cycles of the software must be faster. In addition, the fleet management and integration to ERP systems is already a widely recognized trend in the industry and the machine is provided as a service [4].

We conducted a survey where the experts of the machine control systems domain answered what quality attributes they see important now and in five years. The future aspect is interesting as the business domain is changing to more software-oriented. According to presentation by Rantanen [5] from Sandvik Mining and Construction, more and more of the revenue is generated by software. In addition, Research institute of the Finnish Economy concluded in 2011 report [6] where software development in Finnish industrial companies was studied. The findings were, that in average, 49 per cent of the turnover of the manufacturing companies engaged in SW development was dependent on software and it adds 17 billion euros of value to their products [6].

This paper is organized as follows: in the next chapter we explore the significance of the quality in machine control systems. Next, we introduce our research survey and how it was conducted. After that the results are presented. From this information, some key success factors are extracted for MMC domain. Finally, concluding remarks are given.

Duality of the Software Quality

Software quality can be divided into two categories: internal and external quality [7]. External quality is what interests the end user: if the system has all needed functions, is it easy to use, and does it produce the correct end results and so forth. These are often captured in the functional requirements of the system. The external quality presents all those attributes the customer is willing to pay for. On the other hand, internal quality attributes are interesting for the development team. Investing in the internal quality is making the future work easier if the system evolves as planned. Thus, good internal quality might save lot of development time and subsequently cut costs of adding new marketable features. If the team, as a one stakeholder for the product, has succeeded to capture internal quality in the requirements, they are often described as separate requirements. They are usually solved on architectural level as the internal quality requirements are crosscutting issues which cannot be addressed on feature level. The study by Ameller [2] focuses on the NFRs which are described as quality attribute requirements and constraints. The end results seems to be that the architect is the main source for the internal quality attributes, but the main driver for the development is the functionality and NFRs that are easily perceivable to the user; performance, usability and security. They further remark that sometimes the customer usually present their NFRs as vague requirements or do not mention them at all, but still complain when their presumed quality level has not been attained.

As Buschmann et al. [1] noted that the results from Ameller’s study were surprising and conclude that architects main responsibility is to decide if the internal quality has value. In some cases, a big ball of mud [8] may be an appropriate architecture. However, the architecture has traditionally strong emphasis in MMC domain and the survey was conducted to see if the results apply also in this domain.

The Survey

The survey was carried out as questionnaire handed out to audience of Intelligent Machines day held at the Tampere University of Technology in March 2013. The audience consisted of people from the industry and academia with various backgrounds. The industry participants were mainly from mobile machine manufacturing and process automation companies.

In the survey, the position of the respondent (architect, developer, etc.) was asked and based on that, the answers were grouped into seven separate groups. The main separation criterion was if the respondent was from the industry or from academia. These two groups were further divided into subgroups based on their position. The groups and the number of responses for that particular group are summarized in Table 1. As shown in the table, the largest group was formed by the developers from the industry companies. In addition, the industry category clearly outnumbers the academia category.

Table 1: Participant distribution

Position	Group	N
Academic, architect	AA	2
Academic, developer	AD	8
Academic, others	AO	6
Industry, architect	IA	7
Industry, developer	ID	14
Industry, manager	IM	7
Industry, others	IO	5
Total		49

The survey was constructed as a one paper sheet where we listed issues: the main quality attributes from the ISO 9126 quality standard [9] and in addition the main constraints from Ameller [2]. So, the total number of issues was 28. Moreover, there were a couple of empty lines, if the participant had some important attributes of her own to mention.

The participants were asked to prioritize the five main issues in the domain. They were also asked to consider them now and in five year time span. Each response was scored so that the most important issue marked by the participant was worth of 5 points and the least important one was 1 point. The points were summed up and sorted.

Results

In Figure 1, the results from the survey are illustrated. It shows prioritization results and the total points given by respondents. Table 2 gives a summary on the main issues considered by each group.

To find out, if there are any differences between different roles, the answers were grouped into four groups by combining industry and academia respondents by their roles, namely *architects*, *developers*, *managers*, and *others.*

Table 2: The main issues

Group	Now	5 years
AA	Learnability	Understandability
AD	Costs	Costs
AO	Stability	Costs
IA	Stability/Costs	Security
ID	Stability	Stability
IM	Costs	Changeability
IO	Costs	External regulations/ Simplicity
All	Stability	Security

For architects, two main issues today were costs and stability. Stability of the design and components is important because of the very long life cycle of the automation control systems. Additionally, changes in selected components during development time might increase development time and costs. As the issues for future, architects prioritized interoperability and security as the most important ones. These attributes correspond with the current trend to network different kinds of systems together so that they can communicate information. In mobile machine domain the trend is to create seamless information flow between different kinds of work machines. For example, in forestry domain, the work order is sent from the paper mill to the harvester which fells the logs and informs the coordinates of the location of the logs to forwarder which again might send information forward to log truck.

Developers prioritized stability and costs as main issues for the current systems. The stability will also remain as the main issue in the future, but interoperability will be more important issue than costs. This also indicates the increase of need for fleet management and co-operation of different systems.

From manager point of view, costs are the most relevant issue today. In the future, changeability will be the most important issue reflecting increasing competition and importance of differentiation in MMC domain. The changeability is also important as development speed needs to be increased as the new products must be brought to market with shorter lead time.

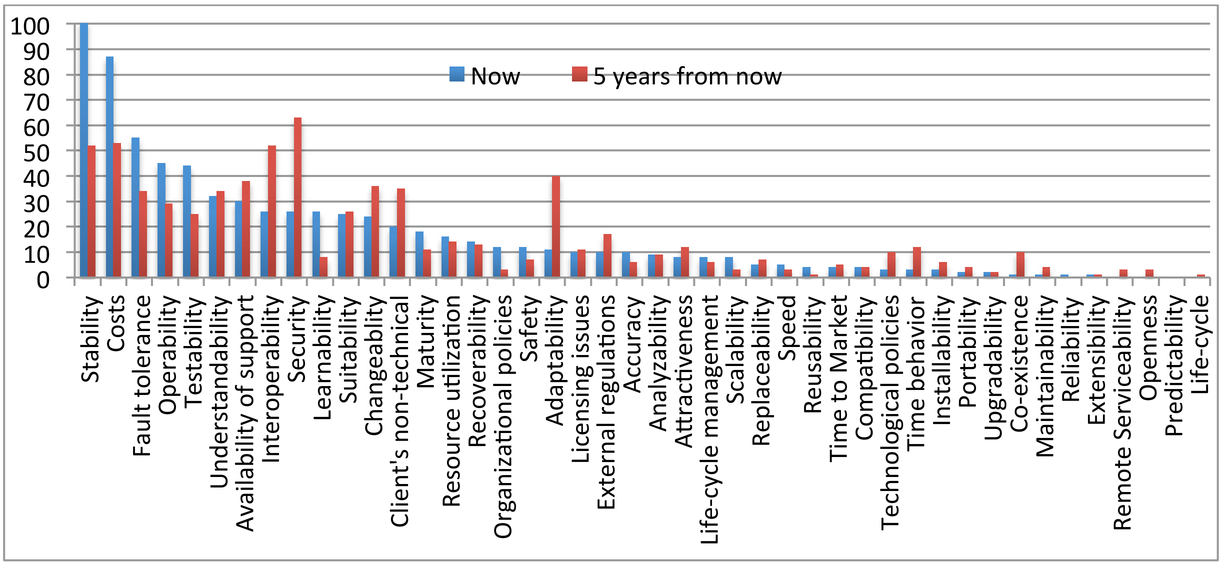

Figure 1. Points given for the issues

The others group had costs and stability as the current issues and changeability and security for the future challenges. These answers were in line with the other groups.

When the results are compared to Ameller's [2] or Svensson's et al. [10], there are several differences. For example, licensing issues were almost totally omitted by our respondents. Likewise, performance seems to be no issue in MMC domain. This might be due the hardware being typically designed specifically for the software in collaboration between hardware team with guidance from the software team. Some of the differences can be explained with the different issues between different studies. On the other hand, licensing issues are often taken care by someone else than the designer or the architect of the system.

In order to make the results between our survey and Ameller's study more comparable, we grouped the quality attributes according to main characteristics in the ISO quality model, see Figure 2 for details. We also compared the constraints between these two studies and the results are shown in Figure 3. The results show that the quality attributes considered traditionally important in the MMC field, such as reliability, usability and maintainability, seem to rise above of other quality attributes. The functionality that was the surprising result in Ameller's study does not show off to have any special importance.

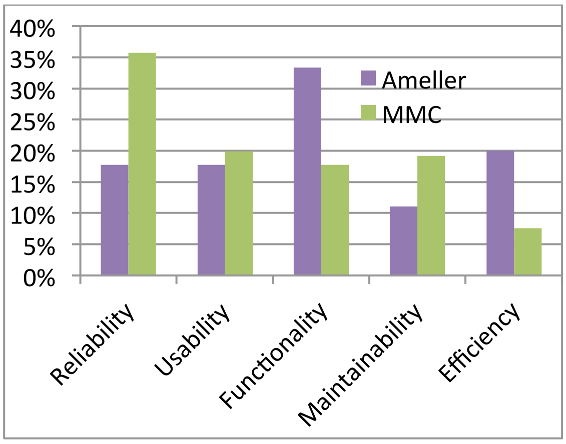

Figure 2. Comparison of main quality attributes

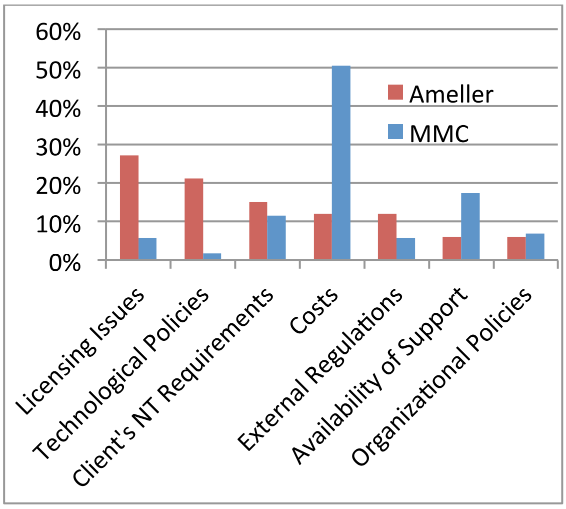

Figure 3. Comparison of constraints

To summarize results from the survey, the domain is moving from the current, relatively stable environment to open, networked environment. For this reason, security and interoperability attributes were considered as top-level issues in the near future. For many groups, especially in the academia category, the cost-efficiency (*costs*) was still the main issue for now and in the future. This finding is in line with what Ameller et al. said before in [2]. Costs are seen as a prominent factor during the system's design time.

Key Success Factors

As the survey results show, cost effectiveness is one of the most critical issues in MMC domain currently. The result might be partly caused by the current economic trend. However, there might be something more deep rooted in that answer.

Optimizing cost effectiveness might be carried out on the expense of other quality attributes, e.g. efficiency. Often, cheaper COTS components are selected and they might have lower processing power. However, this optimizing might lead to local optimizations where costs are saved by selecting low end components, but overall costs are increased as software needs to be changed due the hardware changes and causes additional work for the developers. Consequently, these development costs may generate new needs to save costs. Therefore, it is crucial that the costs are rather optimized on global level than locally. The need for global optimization is also recognized in lean software development approach [11].

Changing hardware and COTS software components also lead to need for stability which was prioritized as one of the most important success factors. From software architecture point of view, it means more decoupling and interfacing which may somewhat decrease the performance. On the other hand, it means that more flexibility is needed from the development and agile methods need to be adopted on the organization level. For example, Eklund et al. [12] have studied how agile methods fit the development of mass-produced embedded systems in automotive industry.

Additionally, achieving stability requires more cross-disciplinary collaboration of different teams. Software designers must work in collaboration with electronics, hydraulics and mechanics designers. However, it is also recognized to be one of the main challenges of agile methods [13] as it means interfacing with non-agile teams.

The survey revealed that the importance of security and interoperability will increase in five years' time. This is quite expected result as mobile machines are becoming more and more networked and communicate with external systems, such as ERP, and with each other. This requires interoperability and as the machines network more, it opens new attack vectors and increases the need for security.

Currently vehicular communication systems have several security problems as pointed out in [14]. Another recent research revealed that there were almost 3000 automation systems that could be accessed by anyone from the internet without any authentication [15]. Additionally, when the machines are networked and vulnerabilities are found from the system, it implies that the software must be updateable easily and frequently with minimal effort. This might require new approaches in the future.

Conclusions

The results show that costs and stability are currently the most important issues in the mobile machine domain according to the survey. In the future as the machines will be more networked, the importance of security and interoperability will grow.

When comparing the results Ameller et al. [2] it seems that reliability and costs are more important on mobile machine domain than in other domains where functionality and licensing issues are more important.

The results presented in this paper will function as a basis for future research on which the key marketing and differentiating points in the control system software are studied.

Veli-Pekka Eloranta

Tampere University of Technology

P.O. Box 553, FIN-33101 Tampere, Finland

+358 40 849 0398

Fax +358 3 3115 2913

veli-pekka.eloranta@tut.fi

http://patterns.cs.tut.fi

Johannes Koskinen
Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland
+358 40 849 0750
Fax +358 3 3115 2913
johannes.koskinen@tut.fi
http://patterns.cs.tut.fi/

Marko Leppänen
Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland
+358 40 849 0387
Fax +358 3 3115 2913
marko.leppanen@tut.fi
http://patterns.cs.tut.fi/

References

[1] F. Buschmann, D. Ameller, C. Ayala, J. Cabot and X. Franch, "Architecture Quality Revisited," IEEE Software, vol. 29, no. 4, pp. 22-24, 2012.

[2] D. Ameller, C. Ayala, J. Cabot and X. Franch, "Non-functional requirements in software architecture practice," Report ESSI-TR-12-1, Universitat Politècnica de Catalunya. Available online: http://hdl.handle.net/2117/15716, Barcelona, 2012.

[3] U. Eklund and J. Bosch, "Introducing Software Ecosystems for Mass-Produced Embedded Systems," in Proceedings of Third International Conference On Software Business, 2012.

[4] J. Glueck, P. Koudal and W. Vaessen, "The service revolution," Deloitte, Report available online: http://www.deloitte.com/view/en_US/us/Insights/Browse-by-Content-Type/deloitte-review/563a4d63e70fb110VgnVCM100000ba42f00aRCRD.htm, 2011.

[5] A. Rantanen, "Intelligent Machines' Software Day," 27 03 2013. [Online]. Available: http://www.tut.fi/fi/tietoa-yliopistosta/laitokset/tietotekniikka/laitoksen-esittely/tapahtumia/viksut-vekottimet/index.htm. [Accessed 17 5 2013].

[6] T. Nikulainen, J. Ali-Yrkkö and T. Seppälä, "Softaa Koneisiin! Ohjelmisto-osaaminen suomalaisen teollisuuden uudistajana," Teknologiateollisuus ry ISBN: 978-952-628-535-4, 2011.

[7] S. McConnell, Code Complete 2nd edition, Microsoft Press ISBN 978-0735619678, 2004.

[8] B. Foote and J. W. Yoder, "Big Ball of Mud," in Pattern Languages of Program Design 4, Addison-Wesley, 2000, pp. 653-692.

[9] ISO/IEC, "9126-1:2001 Software engineering — Product quality — Part 1: Quality model," 2001.

[10] R. Svensson, T. Gorschek, B. Regnell, R. Torkar, A. Shahrokni and R. Feldt, "Quality Requirements in Industrial Practice - An Extended Interview Study at Eleven Companies," IEEE Transactions on Software Engineering, vol. 38, no. 4, pp. 923-935, 2012.

[11] M. Poppendieck and T. Poppendieck, Lean Software Development: An Agile Toolkit, Addison-Wesley, 2003.

[12] U. Eklund and J. Bosch, "Applying Agile Development in Mass-Produced Embedded Systems," in Proceedings of 13th International Conference Agile Software Development, XP 2012, 2012.

[13] R. Baskerville, J. Pries-Heje and S. Madsen, "Post-agility: What follows a decade of agility," Journal of Information and Software Technology, vol. 53, no. 5, pp. 543-555, 2011.

[14] M. Wolf, A. Weimerskirch and C. Paar, "Security in Automotive Bus Systems," in Proceedings of the Workshop on Embedded Security in Cars (ESCAR '04), 2004.

[15] S. Tiilikainen and J. Manner, "Vulnerability of Finnish Automation Networks," Aalto University, 2013.